Algorithmic Graph Minors:

turning Combinatorics to Algorithms

Dimitrios M. Thilikos

Department of Mathematics – $\mu\prod\lambda\forall$

UoA - National and Kapodistrian University of Athens

Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier

(LIRMM)

Montpellier, February 2, 2012

Let $\mathcal{X}$ be a set and let "$\leq$" be a partial ordering relation on $\mathcal{X}$.

Let $\mathcal{X}$ be a set and let "$\leq$" be a partial ordering relation on $\mathcal{X}$.

Antichain: an infinite sequence on non- $\leq$-comparable elements.

Let $\mathcal{X}$ be a set and let "$\leq$" be a partial ordering relation on $\mathcal{X}$.

Antichain: an infinite sequence on non- $\leq$-comparable elements.

We say that $\mathcal{X}$ is *Well-Quasi-Ordered* by $\leq$ if it has no infinite antichain

Let $\mathcal{X}$ be a set and let "$\leq$" be a partial ordering relation on $\mathcal{X}$.

Antichain: an infinite sequence on non- $\leq$-comparable elements.

We say that $\mathcal{X}$ is *Well-Quasi-Ordered* by $\leq$ if it has no infinite antichain

Examples:

- $2^{\mathbb{N}}$ is not W.Q.O. by set inclusion.
- $\mathbb{N}$ is not W.Q.O. by divisibility.
- $\mathbb{N}^k$ is W.Q.O. by by component-wise ordering.

General question: Given a set $\mathcal{X}$ and an ordering relation $\leq$ on it,

is $\mathcal{X}$ W.Q.O. by to $\leq$?

General question: Given a set $\mathcal{X}$ and an ordering relation $\leq$ on it,

is $\mathcal{X}$ W.Q.O. by to $\leq$?

The **theory of Well-quasi-ordering** was first developed by

Graham Higman and Erdős & Rado

under the name "finite basis property"

General question: Given a set $\mathcal{X}$ and an ordering relation $\leq$ on it, is $\mathcal{X}$ W.Q.O. by to $\leq$?

The **theory of Well-quasi-ordering** was first developed by Graham Higman and Erdős & Rado under the name "finite basis property"

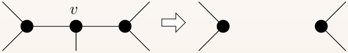
Remind: This talk is about graphs and algorithms!

Graph Minors

We define 3 local operations on graphs:

Graph Minors

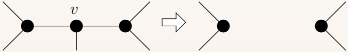
We define 3 local operations on graphs:

1. $\backslash v$: vertex removal: 

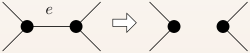
Graph Minors

We define 3 local operations on graphs:

1. $\backslash v$: vertex removal:



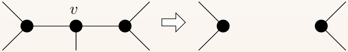
2. $\backslash e$: edge removal:



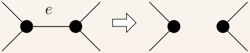
Graph Minors

We define 3 local operations on graphs:

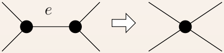
1. $\backslash v$: vertex removal:



2. $\backslash e$: edge removal:



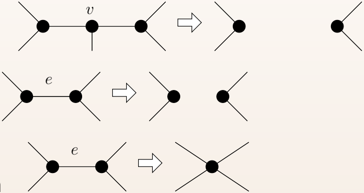
3. $/e$: edge contraction



Graph Minors

We define 3 local operations on graphs:

1. $\backslash v$: vertex removal:
2. $\backslash e$: edge removal:
3. $/e$: edge contraction



Minor Relation:

$H \leq G$ if H can be obtained from G after a sequence of the above operations

Wagner's Conjecture:

- The set of all graphs is W.Q.O. by the minor relation

[formulated by Klaus Wagner in the 1930s (?)]

Wagner's Conjecture:

- The set of all graphs is W.Q.O. by the minor relation

[formulated by Klaus Wagner in the 1930s (?)]

This conjecture was proven by *Neil Robertson* and *Paul Seymour* in their Graph Minor series of papers.

This conjecture was proven by *Neil Robertson* and *Paul Seymour* in their

Graph Minor series of papers.

This conjecture was proven by *Neil Robertson* and *Paul Seymour* in their Graph Minor series of papers.

This conjecture was proven by *Neil Robertson* and *Paul Seymour* in their Graph Minor series of papers.

Now it is known as the *Robertson & Seymour Theorem*.

This conjecture was proven by *Neil Robertson* and *Paul Seymour* in their

Graph Minor series of papers.

Now it is known as the *Robertson & Seymour Theorem*.

Width of the proof: < 10 cm

This conjecture was proven by *Neil Robertson* and *Paul Seymour* in their Graph Minor series of papers.

Now it is known as the *Robertson & Seymour Theorem*.

Width of the proof: < 10 cm (23 papers)

This conjecture was proven by *Neil Robertson* and *Paul Seymour* in their

Graph Minor series of papers.

Now it is known as the *Robertson & Seymour Theorem*.

Width of the proof: < 10 cm (23 papers)

10/11 Fulkerson Prize (2006) (2/3 for N.R. & 4/5 for P.S.)

In this talk we will present ( ) the algorithmic applications of the Graph Minors Series.

In this talk we will present (some of) the algorithmic applications of the Graph Minors Series.

We say that a parameterized (by k) problem belongs in the parameterized complexity class FPT if it can be solved by an FPT-algorithm, that is an algorithm that runs in

$$O(f(k) \cdot n^{O(1)}) \text{ steps}$$

(n is the size of the input, f depends only one the parameter k.)

We say that a parameterized (by k) problem belongs in the parameterized complexity class FPT if it can be solved by an FPT-algorithm, that is an algorithm that runs in

$$O(f(k) \cdot n^{O(1)}) \text{ steps}$$

(n is the size of the input, f depends only one the parameter k.)

- Not all parameterized problems admit FPT-algorithms.

We say that a parameterized (by k) problem belongs in the parameterized complexity class FPT if it can be solved by an FPT-algorithm, that is an algorithm that runs in

$$O(f(k) \cdot n^{O(1)}) \text{ steps}$$

(n is the size of the input, f depends only one the parameter k.)

- Not all parameterized problems admit FPT-algorithms.

There are parameterized complexity classes like W[1], W[2], or W[P] and adequate reductions such that when a problem is hard for them is not expected to have an FPT-algorithm.

graph parameter: a function **p** that maps graphs to integers.

graph parameter: a function **p** that maps graphs to integers.

A **meta**-problem:

k-Parameter **p**-Checking
Instance: a graph G and an integer $k \geq 0$.
Parameter: k
Question: $\mathbf{p}(G) \leq k$?

graph parameter: a function **p** that maps graphs to integers.

A meta-problem:

k-PARAMETER **p**-CHECKING
Instance: a graph G and an integer $k \geq 0$.
Parameter: k
Question: **p**$(G) \leq k$?

p can be the minimum VERTEX COVER, DOMINATING SET, EDGE DOMINATING SET, CHROMATIC NUMBER, FEEDBACK VERTEX SET, e.t.c.

graph parameter: a function **p** that maps graphs to integers.

A meta-problem:

k-PARAMETER **p**-CHECKING
Instance: a graph G and an integer $k \geq 0$.
Parameter: k
Question: **p**(G)$\leq k$?

p can be the minimum VERTEX COVER, DOMINATING SET, EDGE DOMINATING SET, CHROMATIC NUMBER, FEEDBACK VERTEX SET, e.t.c.

- Holy grail (meta)-question

graph parameter: a function **p** that maps graphs to integers.

A **meta**-problem:

k-PARAMETER **p**-CHECKING
Instance: a graph G and an integer $k \geq 0$.
Parameter: k
Question: **p**$(G) \leq k$?

p can be the minimum VERTEX COVER, DOMINATING SET, EDGE DOMINATING SET, CHROMATIC NUMBER, FEEDBACK VERTEX SET, e.t.c.

- Holy grail (meta)-question

For which functions **p** it holds that k-PARAMETER **p**-CHECKING∈FPT?

(i.e., there is an $f(k) \cdot n^{O(1)}$-step algorithm checking whether **p**$(G) \leq k$?)

A parameter **p** is *minor closed* if $H \leq G \Rightarrow \mathbf{p}(H) \leq \mathbf{p}(G)$.

A parameter **p** is *minor closed* if $H \leq G \Rightarrow \mathbf{p}(H) \leq \mathbf{p}(G)$.

Minor-closed parameters:

- vertex cover, $\mathbf{vc}(G)$
- feedback vertex set, $\mathbf{fvs}(G)$
- branchwidth/treewidth/pathwidth/tree-depth,

  $\mathbf{bw}(G)/\mathbf{tw}(G)/\mathbf{pw}(G)/\mathbf{td}(G)$
- minimum maximal matching, $\mathbf{mmm}(G)$
- $\mathbf{p}(G) = |V(G)| - \alpha(G)$ ($\alpha(G)$ is the max independent set size)
- the genus of a graph, $\gamma(G)$
- the apex number of a graph, $\mathbf{apx}(G)$
- $\mathbf{p}(G) = \min\{k \mid P_k \not\leq G\}$ (Longest Path)

Main meta-algorithmic consequence of GMT:

Main meta-algorithmic consequence of GMT:

- If **p** is minor closed then k-PARAMETER **p**-CHECKING $\in$FPT.

Main meta-algorithmic consequence of GMT:

- If **p** is minor closed then k-PARAMETER **p**-CHECKING $\in$FPT.

In other words,

- **p**$(G) \leq k$ can be checked in $f(k) \cdot n^3$ steps.
- Every minor-closed graph class $\mathcal{G}$ can be recognized in $O(n^3)$.

  [Take **p**$(G) = 0$ if $G \in \mathcal{G}$ and **p**$(G) = 1$, otherwise]

Questions on the last two versions of the meta-algorithmic result:

Questions on the last two versions of the meta-algorithmic result:

- $\mathbf{p}(G) \leq k$ can be checked in $f(k) \cdot n^3$ steps.

Questions on the last two versions of the meta-algorithmic result:

- $\mathbf{p}(G) \leq k$ can be checked in $f(k) \cdot n^3$ steps.

What is f?

Questions on the last two versions of the meta-algorithmic result:

- $\mathbf{p}(G) \leq k$ can be checked in $f(k) \cdot n^3$ steps.

What is f?

- Every minor-closed graph class $\mathcal{G}$ can be recognized in $O(n^3)$.

Questions on the last two versions of the meta-algorithmic result:

- $\mathbf{p}(G) \leq k$ can be checked in $f(k) \cdot n^3$ steps.

What is f?

- Every minor-closed graph class $\mathcal{G}$ can be recognized in $O(n^3)$.

What is hidden in the O-notation?

Questions on the last two versions of the meta-algorithmic result:

- $\mathbf{p}(G) \leq k$ can be checked in $f(k) \cdot n^3$ steps.

What is f?

- Every minor-closed graph class $\mathcal{G}$ can be recognized in $O(n^3)$.

What is hidden in the O-notation?

☠-question: Is there any practical algorithm here?

Questions on the last two versions of the meta-algorithmic result:

- $\mathbf{p}(G) \leq k$ can be checked in $f(k) \cdot n^3$ steps.

What is f?

- Every minor-closed graph class $\mathcal{G}$ can be recognized in $O(n^3)$.

What is hidden in the O-notation?

☠-question: Is there any practical algorithm here?

... go back to the proofs!

- For any minor closed parameter $\mathbf{p}$ and any k, we define $\mathbf{ob}_k(\mathbf{p})$ as the set of minor-minimal elements in

$$\{G \mid \mathbf{p}(G) > k\}$$

- we call $\mathbf{ob}_k(\mathbf{p})$ obstruction family of $\mathbf{p}$.
- Observe: $\mathbf{p}(G) \leq k \Leftrightarrow \forall_{H \in \mathbf{ob}_k(\mathbf{p})}\ H \not\leq G$
- Observe: $\mathbf{ob}_k(\mathbf{p})$ is an antichain.
- GMT Consequence: $\mathbf{ob}_k(\mathbf{p})$ is finite!

An algorithm for the k-PARAMETER **p**-CHECKING problem

1. for all $H \in \mathbf{ob}_k(\mathbf{p})$
2. check (in $O(g(k) \cdot n^3)$ steps) whether $H \leq_{mn} G$
3. and if this holds, then output NO
4. output YES.

An algorithm for the k-PARAMETER **p**-CHECKING problem

1. for all $H \in \mathbf{ob}_k(\mathbf{p})$
2. check (in $O(g(k) \cdot n^3)$ steps) whether $H \leq_{mn} G$
3. and if this holds, then output NO
4. output YES.

The meta-problem is reduced to the k-MINOR CONTAINMENT problem.

An algorithm for the k-Parameter **p**-Checking problem

1. for all $H \in \mathbf{ob}_k(\mathbf{p})$
2. check (in $O(g(k) \cdot n^3)$ steps) whether $H \leq_{mn} G$
3. and if this holds, then output NO
4. output YES.

The meta-problem is reduced to the k-Minor Containment problem.

k-Minor Containment problem can be solved in $O(g(k) \cdot n^3)$ steps

An algorithm for the k-PARAMETER **p**-CHECKING problem

1. for all $H \in \mathbf{ob}_k(\mathbf{p})$
2. check (in $O(g(k) \cdot n^3)$ steps) whether $H \leq_{mn} G$
3. and if this holds, then output NO
4. output YES.

The meta-problem is reduced to the k-MINOR CONTAINMENT problem.

k-MINOR CONTAINMENT problem can be solved in $O(g(k) \cdot n^3)$ steps

The whole algorithm takes $O(|\mathbf{ob}_k(\mathbf{p})| \cdot g(k) \cdot n^3)$ steps.

An algorithm for the k-PARAMETER **p**-CHECKING problem

1. for all $H \in \mathbf{ob}_k(\mathbf{p})$
2. check (in $O(g(k) \cdot n^3)$ steps) whether $H \leq_{mn} G$
3. and if this holds, then output NO
4. output YES.

The meta-problem is reduced to the k-MINOR CONTAINMENT problem.

k-MINOR CONTAINMENT problem can be solved in $O(g(k) \cdot n^3)$ steps

The whole algorithm takes $O(|\mathbf{ob}_k(\mathbf{p})| \cdot g(k) \cdot n^3)$ steps.

Good news: $g(k)$ is constructible!

An algorithm for the k-PARAMETER **p**-CHECKING problem

1. for all $H \in \mathbf{ob}_k(\mathbf{p})$
2. check (in $O(g(k) \cdot n^3)$ steps) whether $H \leq_{mn} G$
3. and if this holds, then output NO
4. output YES.

The meta-problem is reduced to the k-MINOR CONTAINMENT problem.

k-MINOR CONTAINMENT problem can be solved in $O(g(k) \cdot n^3)$ steps

The whole algorithm takes $O(|\mathbf{ob}_k(\mathbf{p})| \cdot g(k) \cdot n^3)$ steps.

Good news: $g(k)$ is constructible!

-news: $g(k)$ is **huge**!

☠☠-facts on the main meta-algorithmic result of GMT.

☠☠-facts on the main meta-algorithmic result of GMT.

1. the above algorithm "exists" but cannot be constructed as we do not know $\mathbf{ob}_k(\mathbf{p})$

☠☠-facts on the main meta-algorithmic result of GMT.

1. the above algorithm "exists" but cannot be constructed as we do not know $\mathbf{ob}_k(\mathbf{p})$

   ▶ There is no TM that, given a machine description of **p**, can produce $\mathbf{ob}_k(\mathbf{p})$. [Fellows & Langston, JCSS 1994]

☠☠-facts on the main meta-algorithmic result of GMT.

1. the above algorithm "exists" but cannot be constructed as we do not know $\mathbf{ob}_k(\mathbf{p})$
   - There is no TM that, given a machine description of **p**, can produce $\mathbf{ob}_k(\mathbf{p})$. [Fellows & Langston, JCSS 1994]
2. we know $\mathbf{ob}_k(\mathbf{p})$ for few parameters and for small values of k

☠☠-facts on the main meta-algorithmic result of GMT.

1. the above algorithm "exists" but cannot be constructed as we do not know $\mathbf{ob}_k(\mathbf{p})$
   - There is no TM that, given a machine description of **p**, can produce $\mathbf{ob}_k(\mathbf{p})$. [Fellows & Langston, JCSS 1994]
2. we know $\mathbf{ob}_k(\mathbf{p})$ for few parameters and for small values of k
3. when we have upper bounds for $|\mathbf{ob}_k(\mathbf{p})|$, they are immense.

Robertson & Seymour, proved the following:

Theorem (GM-VI, GM-VII, GM-XIII, GM-XXI, GM-XII)

The following two problems can be solved in $O(g(k) \cdot n^3)$ steps:

Robertson & Seymour, proved the following:

Theorem (GM-VI, GM-VII, GM-XIII, GM-XXI, GM-XII)

The following two problems can be solved in $O(g(k) \cdot n^3)$ steps:

k-Minor Containment
Instance: two graphs G and H.
Parameter: $k = \|V(H)\|$
Question: $H \leq G$?

Robertson & Seymour, proved the following:

Theorem (GM-VI, GM-VII, GM-XIII, GM-XXI, GM-XII)

The following two problems can be solved in $O(g(k)\cdot n^3)$ steps:

k-MINOR CONTAINMENT

Instance: two graphs G and H.

Parameter: $k=|V(H)|$

Question: $H \leq G$?

k-DISJOINT PATHS

Instance: A graph G and a sequence of pairs of terminals

$$T=(s_1,t_1),\dots,(s_k,t_k)\in (V(G)\times V(G))^k.$$

Parameter: k.

Question: Are there k pairwise vertex disjoint paths $P_1,\dots,P_k$ in G such that for every $i\in\{1,\dots,k\}$, P_i has endpoints s_i and t_i?

Both k-MINOR CONTAINMENT and k-DISJOINT PATHS PROBLEM where solved using the

The irrelevant vertex Technique

introduced in [GM XIII]

Given an instance (G, T, k) of the k-Disjoint Paths problem,

a vertex $v \in V(G)$ is an *irrelevant* vertex of G if

(G, T, k) and $(G \setminus v, T, k)$ are equivalent instances of the problem.

Given an instance (G, T, k) of the k-Disjoint Paths problem,

a vertex $v \in V(G)$ is an *irrelevant* vertex of G if

(G, T, k) and $(G \setminus v, T, k)$ are equivalent instances of the problem.

Idea: ▶ Find irrelevant vertex and recurse!

Given an instance (G, T, k) of the k-Disjoint Paths problem,

a vertex $v \in V(G)$ is an *irrelevant* vertex of G if

(G, T, k) and $(G \setminus v, T, k)$ are equivalent instances of the problem.

Idea: ▶ Find irrelevant vertex and recurse!

We give a outline of the idea for the case of k-Disjoint Paths.

The general scheme of the algorithm in [GM XIII] is the following:

Input: An instance (G, T, k) of k-Disjoint Paths

Output: An equivalent instance (G, T, k) k-Disjoint Paths

1. **while** $G \notin \mathcal{G}_k$,
2. find an irrelevant vertex v in G
3. set $G \leftarrow G \setminus v$
4. **output** (G, T, k)

The general scheme of the algorithm in [GM XIII] is the following:

Input: An instance (G, T, k) of k-Disjoint Paths

Output: An equivalent instance (G, T, k) k-Disjoint Paths

1. **while** $G \notin \mathcal{G}_k$,
2. find an irrelevant vertex v in G
3. set $G \leftarrow G \setminus v$
4. **output** (G, T, k)

Here $\mathcal{G}_k$ represents some structural condition for the problem input.

The algorithmic scheme depends on the parameterized class $\mathcal{G}_k$ and creates an equivalent instance that belongs in $\mathcal{G}_k$.

- It is applied first for

$$\mathcal{G}_k = \{G \mid G \text{ is a } K_{h(k)}\text{-minor free graph}\}$$

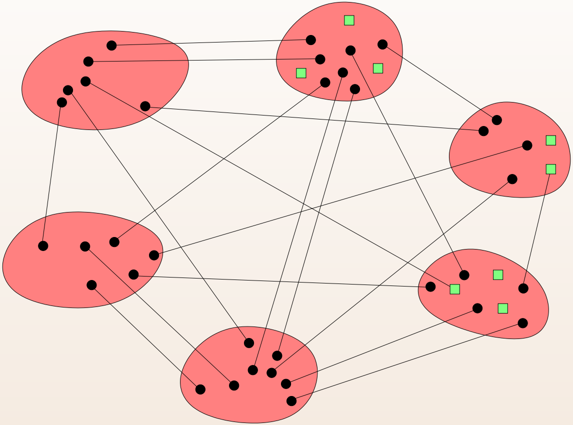
and then for

$$\mathcal{G}_k = \{G \mid G \text{ is a } \Gamma_{j(k)}\text{-minor free graph}\},$$

for some suitable choice of recursive functions $h, j : \mathbb{N} \to \mathbb{N}$.

- Γ_r is the $(r \times r)$-grid.

Phase 1: What to do with a "big" *clique minor*?



(technical) details omitted...

Assume now that the input graph excludes the clique $K_{h(k)}$ as a minor.

Combinatorial question: How such graphs look like?

Assume now that the input graph excludes the clique $K_{h(k)}$ as a minor.

Combinatorial question: How such graphs look like?

Two answers:

Assume now that the input graph excludes the clique $K_{h(k)}$ as a minor.

Combinatorial question: How such graphs look like?

Two answers:

- Weak Structure Theorem [GM XIII]

Assume now that the input graph excludes the clique $K_{h(k)}$ as a minor.

Combinatorial question: How such graphs look like?

Two answers:

- Weak Structure Theorem [GM XIII]
- Strong Structure Theorem [GM XVI]

Theorem (Weak Structure Theorem)

There exists recursive functions $g_1 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ and $g_2 : \mathbb{N} \to \mathbb{N}$, such that for every graph G and every $r, q \in \mathbb{N}$, one of the following holds:

1. K_r *is a minor of* G,
2. $\Gamma_{g_1(r,q)}$ *is not a minor of* G,
3. $\exists X \subseteq V(G)$ *with* $|X| \leq g_2(r)$ *such that* $G \setminus X$ *contains as a subgraph a flat subdivided wall* W *where* W *has height* q

Theorem (Weak Structure Theorem)

There exists recursive functions $g_1 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ and $g_2 : \mathbb{N} \to \mathbb{N}$, such that for every graph G and every $r, q \in \mathbb{N}$, one of the following holds:

1. K_r *is a minor of* G,
2. $\Gamma_{g_1(r,q)}$ *is not a minor of* G,
3. $\exists X \subseteq V(G)$ *with* $|X| \leq g_2(r)$ *such that* $G \setminus X$ *contains as a subgraph a flat subdivided wall* W *where* W *has height* q *and the compass of* W *has a rural division* $\mathcal{D}$ *such that each internal flap of* $\mathcal{D}$ *has treewidth at most* $g_1(r, q)$.

Theorem (Weak Structure Theorem)

There exists recursive functions $g_1 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *and* $g_2 : \mathbb{N} \to \mathbb{N}$, *such that for every graph* G *and every* $r, q \in \mathbb{N}$, *one of the following holds:*

1. K_r *is a minor of* G, (*This is now excluded for* $r = h(k)$)
2. $\Gamma_{g_1(r,q)}$ *is not a minor of* G,
3. $\exists X \subseteq V(G)$ *with* $|X| \leq g_2(r)$ *such that* $G \setminus X$ *contains as a subgraph a flat subdivided wall* W *where* W *has height* q *and the compass of* W *has a rural division* $\mathcal{D}$ *such that each internal flap of* $\mathcal{D}$ *has treewidth at most* $g_1(r, q)$.

Theorem (Weak Structure Theorem)

There exists recursive functions $g_1 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *and* $g_2 : \mathbb{N} \to \mathbb{N}$*, such that for every graph* G *and every* $r, q \in \mathbb{N}$*, one of the following holds:*

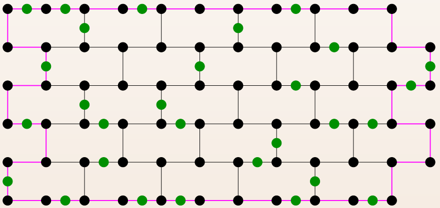
1. K_r *is a minor of* G*, (This is now excluded for* $r = h(k)$*)*
2. $\Gamma_{g_1(r,q)}$ *is not a minor of* G*, (This is what we want to achieve!)*
3. $\exists X \subseteq V(G)$ *with* $|X| \leq g_2(r)$ *such that* $G \setminus X$ *contains as a subgraph a flat subdivided wall* W *where* W *has height* q *and the compass of* W *has a rural division* $\mathcal{D}$ *such that each internal flap of* $\mathcal{D}$ *has treewidth at most* $g_1(r, q)$.

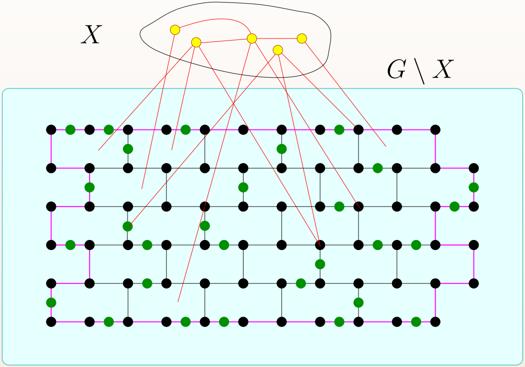
Theorem (Weak Structure Theorem)

There exists recursive functions $g_1 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ and $g_2 : \mathbb{N} \to \mathbb{N}$, such that for every graph G and every $r, q \in \mathbb{N}$, one of the following holds:

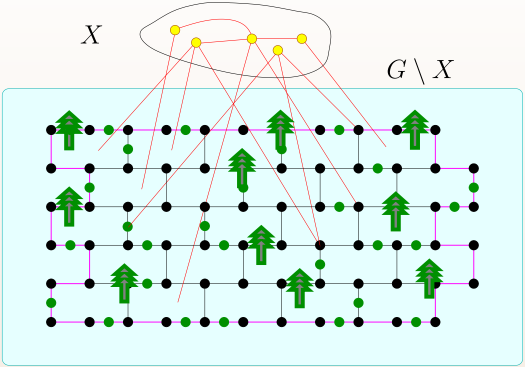
1. K_r *is a minor of* G, *(This is now excluded for $r = h(k)$)*
2. $\Gamma_{g_1(r,q)}$ *is not a minor of* G, *(This is what we want to achieve!)*
3. $\exists X \subseteq V(G)$ *with* $|X| \leq g_2(r)$ *such that* $G \setminus X$ *contains as a subgraph a flat subdivided wall* W *where* W *has height* q *and the compass of* W *has a rural division* $\mathcal{D}$ *such that each internal flap of* $\mathcal{D}$ *has treewidth at most* $g_1(r, q)$.*(irrelevant vertex wanted...)*

A subdivided Wall W of heigh 5:





The compass is the part of the $G \setminus X$ that is "inside" the perimeter of the subdivided wall W. The perimeter is as a separator between the internal compass vertices and the part of $G \setminus X$ that is outside the perimeter



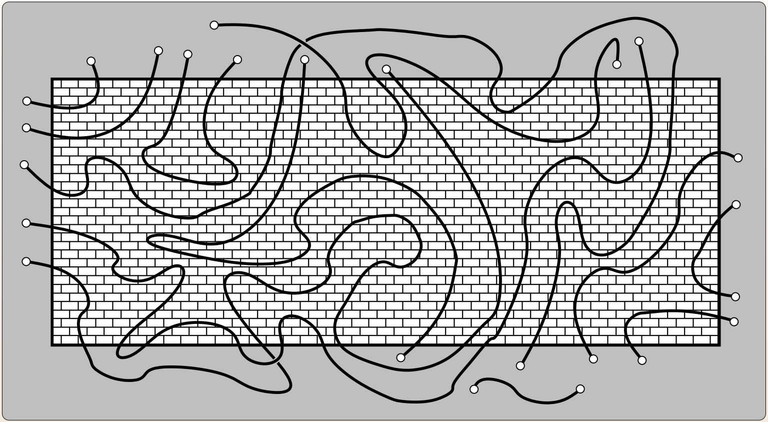
The compass can be decomposed to graphs of bounded treewidth (flaps) whose "roots" have size ≤ 3 and form a planar hypergraph inside the disk bounded by the perimeter

Weak structure theorem $\longrightarrow$ sunny forest theorem!

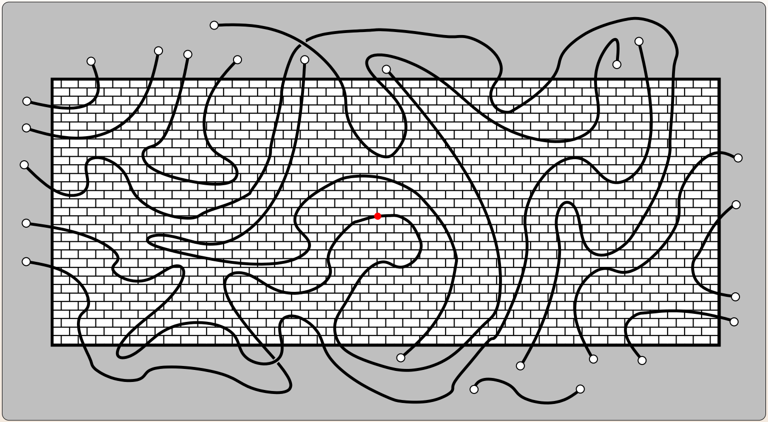
We examine only the simpler case where $X = \emptyset$.

[the forest is **dark**!]

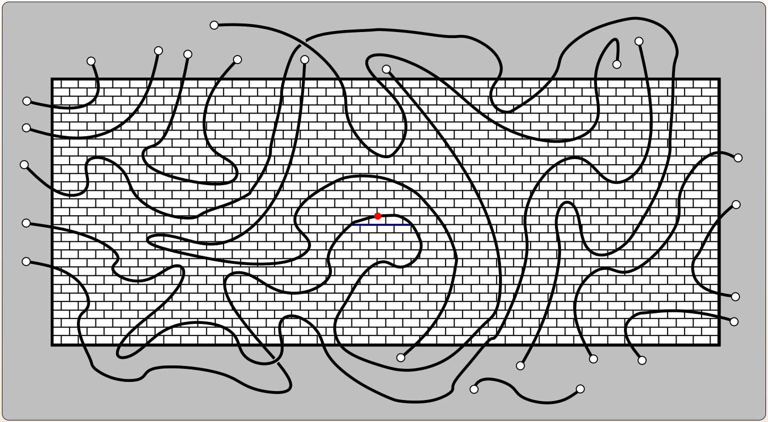
A solution to the k-Disjoint Paths Problem, for $k = 12$



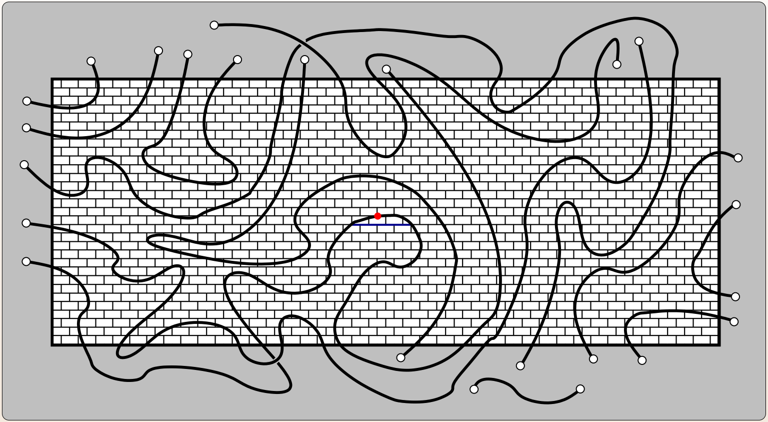
The middle vertex of the subdivided wall W



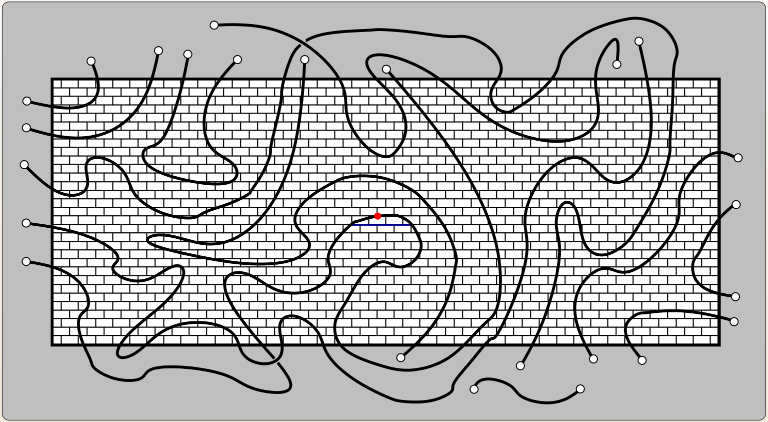
A way to avoid the middle vertex



Is it always possible to avoid the middle vertex?



The answer is YES given that the height of W is at least $\lambda(k)$!



Therefore, if the height of W is "big enough", then we can safely detect an irrelevant vertex (and remove it)!

Theorem (Weak Structure Theorem)

There exists recursive functions $g_1 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ and $g_2 : \mathbb{N} \to \mathbb{N}$, such that for every graph G and every $r, q \in \mathbb{N}$, one of the following holds:

1. K_r *is a minor of* G, *(This is now excluded for $r = h(k)$)*
2. $\Gamma_{g_1(r,q)}$ *is not a minor of* G, *(This is what we want to achieve!)*
3. $\exists X \subseteq V(G)$ *with* $|X| \leq g_2(r)$ *such that* $G \setminus X$ *contains as a subgraph a flat subdivided wall* W *where* W *has height* q *z and the compass of* W *has a rural division* $\mathcal{D}$ *such that each internal flap of* $\mathcal{D}$ *has treewidth at most* $g_1(r,q)$. *(irrelevant vertex wanted...)*

Theorem (Weak Structure Theorem)

There exists recursive functions $g_1 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ and $g_2 : \mathbb{N} \to \mathbb{N}$, such that for every graph G and every $r, q \in \mathbb{N}$, one of the following holds:

1. K_r *is a minor of* G, *(This is now excluded for $r = h(k)$)*
2. $\Gamma_{g_1(r,q)}$ *is not a minor of* G, *(This is what we want to achieve!)*
3. $\exists X \subseteq V(G)$ *with* $|X| \leq g_2(r)$ *such that* $G \setminus X$ *contains as a subgraph a flat subdivided wall* W *where* W *has height* q *z and the compass of* W *has a rural division* $\mathcal{D}$ *such that each internal flap of* $\mathcal{D}$ *has treewidth at most* $g_1(r, q)$. *(irrelevant vertex wanted...)*

- After the second phase, we have an equivalent instance satisfying cond. 2.

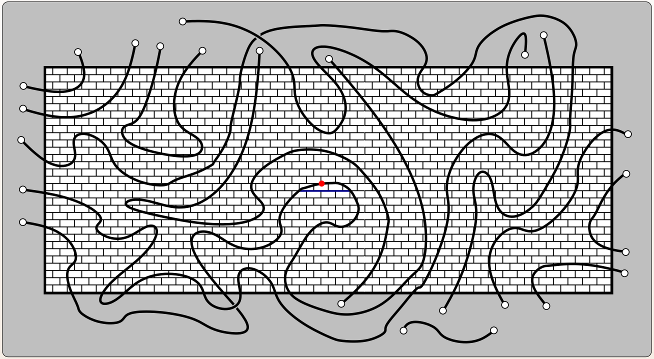
Theorem (Weak Structure Theorem)

There exists recursive functions $g_1 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ and $g_2 : \mathbb{N} \to \mathbb{N}$, such that for every graph G and every $r, q \in \mathbb{N}$, one of the following holds:

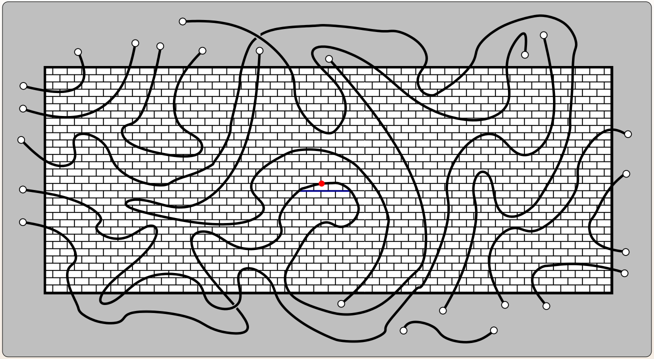
1. K_r *is a minor of* G, *(This is now excluded for $r = h(k)$)*
2. $\Gamma_{g_1(r,q)}$ *is not a minor of* G, *(This is what we want to achieve!)*
3. $\exists X \subseteq V(G)$ *with* $|X| \leq g_2(r)$ *such that* $G \setminus X$ *contains as a subgraph a flat subdivided wall* W *where* W *has height* q *z and the compass of* W *has a rural division* $\mathcal{D}$ *such that each internal flap of* $\mathcal{D}$ *has treewidth at most* $g_1(r, q)$. *(irrelevant vertex wanted...)*

- After the second phase, we have an equivalent instance satisfying cond. 2.
- This means that G has treewidth bounded by some function of k: the problem can be solved using dynamic programming.

We can reroute the k disjoint path, given that the height of W is at least $\lambda(k)$!

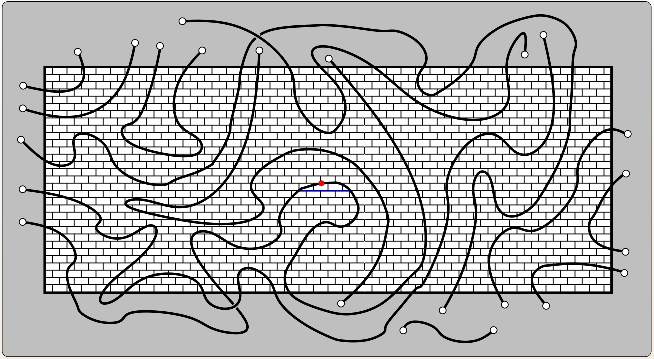


We can reroute the k disjoint path, given that the height of W is at least $\lambda(k)$!



- Proved in [GM XXI].

We can reroute the k disjoint path, given that the height of W is at least $\lambda(k)$!

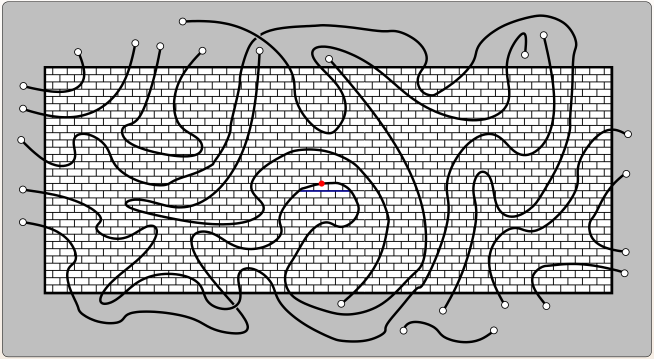


- Proved in [GM XXI].
- The proof uses the "*Vital Linkage Lemma*" (proved in [GM XXI]) and

the "*Strong Structural Theorem of GMT*" (proved in [GM XVI])

☠ What is the estimation of $\lambda(k)$?

We can reroute the k disjoint path, given that the height of W is at least $\lambda(k)$!



- Proved in [GM XXI].
- The proof uses the "*Vital Linkage Lemma*" (proved in [GM XXI]) and the "*Strong Structural Theorem of GMT*" (proved in [GM XVI])

☠ What is the estimation of $\lambda(k)$? Huge!

David Johnson mentioned in his ongoing guide on NP-completeness:

> *"for any instance $G = (V, E)$ that one could fit into the known universe, one would easily prefer $|V|^{70}$ to even constant time, if that constant had to be one of Robertson and Seymour's"*.

David Johnson mentioned in his ongoing guide on NP-completeness:

> *"for any instance $G = (V, E)$ that one could fit into the known universe, one would easily prefer $|V|^{70}$ to even constant time, if that constant had to be one of Robertson and Seymour's".*

David Johnson also estimated that just one constant in the parametric dependence of the strong structural Theorem to be roughly

$$2^{\uparrow 2^{2^{2^{2^{\uparrow 2^{\uparrow \Theta(r)}}}}}}$$

where $2^{\uparrow r}$ denotes a tower $2^{2^{2^{\cdot^{\cdot^{\cdot}}}}}$ involving r 2's.

Improvements!

Improvements!

Target 1: Prove that we can reroute the k disjoint paths without using the "*Strong Structural Theorem of GMT*".

Improvements!

Target 1: Prove that we can reroute the k disjoint paths without using the "*Strong Structural Theorem of GMT*".

Target 2: Find an alternative proof of the "*Strong Structural Theorem of GMT*" that has "better constants".

Improvements!

Target 1: Prove that we can reroute the k disjoint paths without using the "*Strong Structural Theorem of GMT*".

Achieved! By [Ken-ichi Kawarabayashi, Paul Wollan]: A shorter proof of the graph minor algorithm: the unique linkage theorem. STOC 2010

Target 2: Find an alternative proof of the "*Strong Structural Theorem of GMT*" that has "better constants".

Improvements!

Target 1: Prove that we can reroute the k disjoint paths without using the "*Strong Structural Theorem of GMT*".

Achieved! By [Ken-ichi Kawarabayashi, Paul Wollan]: A shorter proof of the graph minor algorithm: the unique linkage theorem. STOC 2010

Target 2: Find an alternative proof of the "*Strong Structural Theorem of GMT*" that has "better constants".

Achieved! By [Ken-ichi Kawarabayashi, Paul Wollan: A simpler algorithm and shorter proof for the graph minor decomposition. STOC 2011]

Improvements!

Target 1: Prove that we can reroute the k disjoint paths without using the "*Strong Structural Theorem of GMT*".

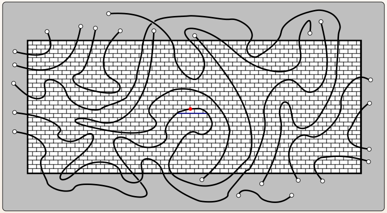
Achieved! By [Ken-ichi Kawarabayashi, Paul Wollan]: A shorter proof of the graph minor algorithm: the unique linkage theorem. STOC 2010

Target 2: Find an alternative proof of the "*Strong Structural Theorem of GMT*" that has "better constants".

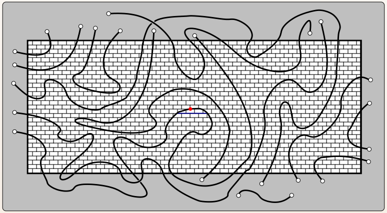
Achieved! By [Ken-ichi Kawarabayashi, Paul Wollan: A simpler algorithm and shorter proof for the graph minor decomposition. STOC 2011]

On going work by: [Reed, Li, Ken-ichi Kawarabayashi]: On going work

Fact: [Ken-ichi Kawarabayashi, Paul Wollan, STOC 2010]: Gives an estimation of $\lambda(k) = 2^{2^{2^{k}}}$ in general gaphs!



Fact: [Ken-ichi Kawarabayashi, Paul Wollan, STOC 2010]: Gives an estimation of $\lambda(k) = 2^{2^{2^{k}}}$ in general gaphs!



This, still, gives an algorithm of $2^{2^{2^{2^{\Omega(k)}}}}$ steps

for the k-Disjoint Paths Problem

Can we make things better?

Can we make things better?

Planar graphs: By [Adler, Kolliopoulos, Krause, Lokshtanov, Saurabh, Thilikos: Tight Bounds for Linkages in Planar Graphs. ICALP 2011] it follows that $\lambda(k) = 2^{O(k)}$.

Can we make things better?

Planar graphs: By [Adler, Kolliopoulos, Krause, Lokshtanov, Saurabh, Thilikos: Tight Bounds for Linkages in Planar Graphs. ICALP 2011] it follows that $\lambda(k) = 2^{O(k)}$.

This gives an $\boxed{2^{2^{\Omega(k)}} \cdot n^3}$ algorithm for the k-Disjoint Paths Problem

Can we make things better?

Planar graphs: By [Adler, Kolliopoulos, Krause, Lokshtanov, Saurabh, Thilikos: Tight Bounds for Linkages in Planar Graphs. ICALP 2011]

it follows that $\lambda(k) = 2^{O(k)}$.

This gives an $\boxed{2^{2^{\Omega(k)}} \cdot n^3}$ algorithm for the k-Disjoint Paths Problem

☠-Limits of the irrelevant vertex technique:

$\boxed{2^{2^{O(k)}} \cdot n^3}$ steps is optimal as $\lambda(k) = 2^{\Omega(k)}$

Can we make things better?

Planar graphs: By [Adler, Kolliopoulos, Krause, Lokshtanov, Saurabh, Thilikos: Tight Bounds for Linkages in Planar Graphs. ICALP 2011] it follows that $\lambda(k) = 2^{O(k)}$.

This gives an $\boxed{2^{2^{\Omega(k)}} \cdot n^3}$ algorithm for the k-Disjoint Paths Problem

☠-Limits of the irrelevant vertex technique:
$\boxed{2^{2^{O(k)}} \cdot n^3}$ steps is optimal as $\lambda(k) = 2^{\Omega(k)}$

- A non-☠ algorithm for the k-Disjoint Paths Problem (and related problems) would require radically different techniques!

Last words on Algorithmic Graph Minors Theory...

Some **recent** FPT-Algorithms using the *irrelevant vertex technique* or variants of it:

- Bipartite Contraction, Partial Vertex Cover, Partial Dominating Set,
- Topological Minor Containment, Immersion Containment,
- Bounded Genus Contraction Containment, Odd Cycle Induced Packing,
- Odd Cycle Packing, Induced cycle, Optimal Embedding in a Surface

Piet Mondrian, Composition with Yellow, Blue, and Red, 1921